Data Mining
ID2222

Homework 5: K-way Graph Partitioning Using JA-BE-JA

Authors:
Abdul Aziz Alkathiri
Francesco Staccone

December 10, 2019

1 Introduction

In this assignment, we implemented the JA-BE-JA (Rahimian et al., 2013) algorithm, which is a distributed algorithm for graph partitioning, on a single node simulation, as well as tweak its different configurations to find the smallest edge cuts for the graphs.

The graph datasets used to complete the tasks are the given datasets, namely the **3elt**, **add20**, and **Facebook** and the **Twitter** graphs.

The datasets are a list of lines that represent the partitions of nodes in a graph, as exemplified in the **3elt** graph dataset shown in Figure 1. The first two numbers in the first line signify the number of partitions and the number of global nodes in the graph respectively.

```
4720 13722
 2 5 4
 3 7 5 1
 6 9 7 2
 1 5 10 15 8
 2 7 11 10 4 1
 12 14 9 3
 3 9 13 11 5 2
 4 15 23 18
 6 14 16 13 7 3
 5 11 17 26 15 4
 7 13 19 17 10 5
 29 22 14 6
 9 16 20 19 11 7
 6 12 22 21 16 9
 4 10 26 34 23 8
 9 14 21 24 20 13
 11 19 30 31 26 10
 8 23 183 327
 13 20 25 27 30 17 11
 16 24 28 25 19 13
 14 22 39 33 24 16
 12 29 46 39 21 14
```

Figure 1: The 3elt graph dataset.

Our submission consists of files and folders of the scaffolding source code for the JA-BE-JA simulation that is given, with the file `Jabeja.java` modified to the completion of the tasks.

We also included two shell script files, `runAlg.sh`, which will run the algorithm on all three graph datasets, and `runPlot.sh`, which will plot the edge-cut, swaps, migrations over the rounds.

2 Implementation and results

2.1 Task 1: JA-BE-JA implementation

For the first task, we implemented the two main tasks of the JA-BE-JA algorithm as described in the paper, that is, `SampleAndSwap` and `FindPartner`.

In `SampleAndSwap`, we find the partner for a node p and swap the colors with its *partner*, which

is one of its neighbors. This is done by first letting every node consider its direct neighbors as candidates for color exchange - this is called *Local policy*. If it fails, every node selects a uniform random sample of nodes from the graph (*Random policy*). In either case, the partner for node p (denoted as node q) is obtained through `FindPartner` by passing the sample nodes (either by Local or Random policy). When a partner is found for node p, they swap colors, and the entire graph is updated accordingly.

`FindPartner` finds the "best partner" (node q) for node p by analyzing each node of the sample. Nodes p and q only exchange colors only if it decreases their *energy* (thus minimizing the *energy of the whole graph*), i.e. increases the number of neighbors with a similar color to that of the node. In the paper, *energy* is defined as the number of edges between nodes with different colors. In other words, the *energy* of a node is the number of neighbors with different colors. The *energy* of the graph is the sum of the *energy* of the nodes, which is what is desired to be minimized.

For each of the graphs, the algorithms ran for 1000 rounds. Figures 2, 4, 6 and 8 respectively show the result for the **3elt**, **add20**, **Twitter** and **Facebook** graphs. Respectively, the corresponding edge-cut, number of swaps and migrations for each graph are shown in Figures 3, 5,7 and 9.

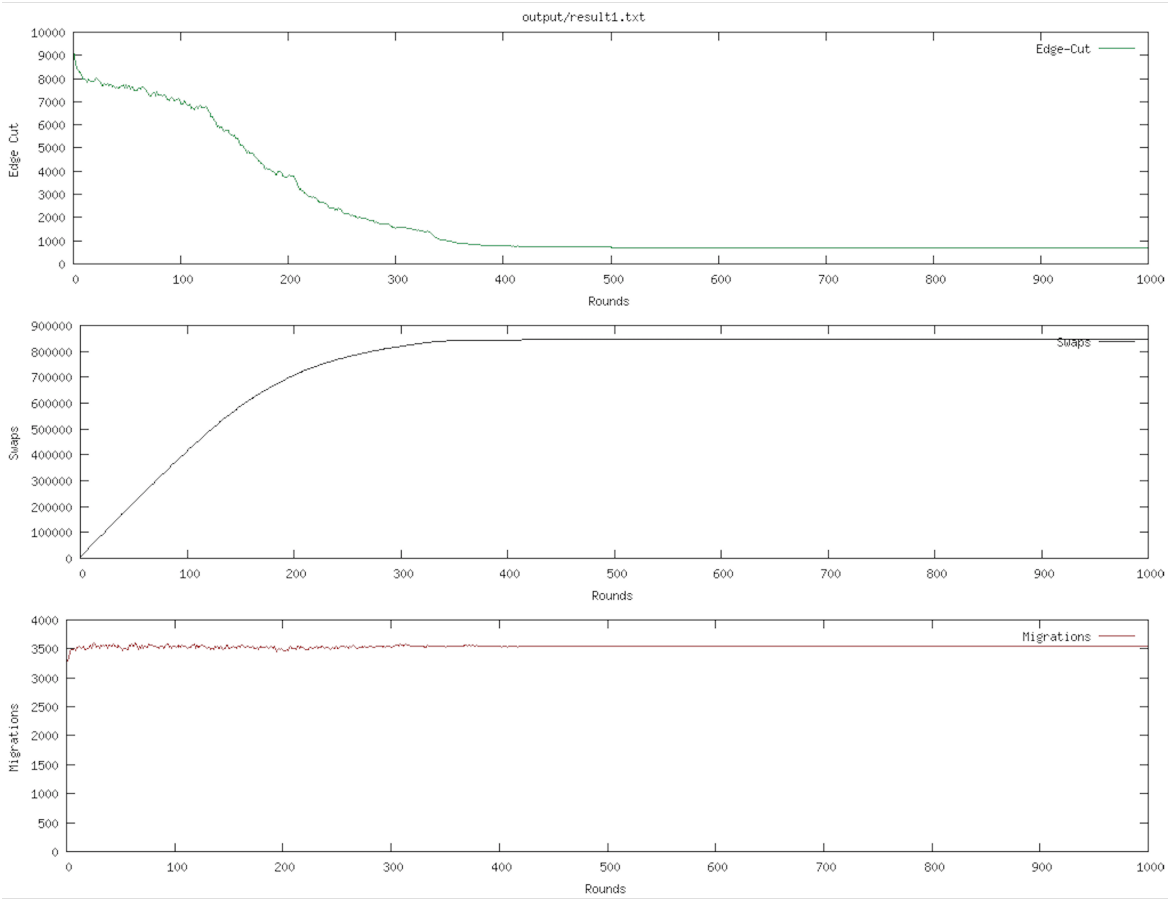

Figure 2: Results of the 3elt graph.

```
2019-12-07 18:20:54 INFO  Jabeja:297 - round: 999, edge cut:711, swaps: 845631, mig
rations: 3544
Minimum edge cut: 711.0
Mac:id2222-master francescostaccone$
```

Figure 3: Stats of the 3elt graph.

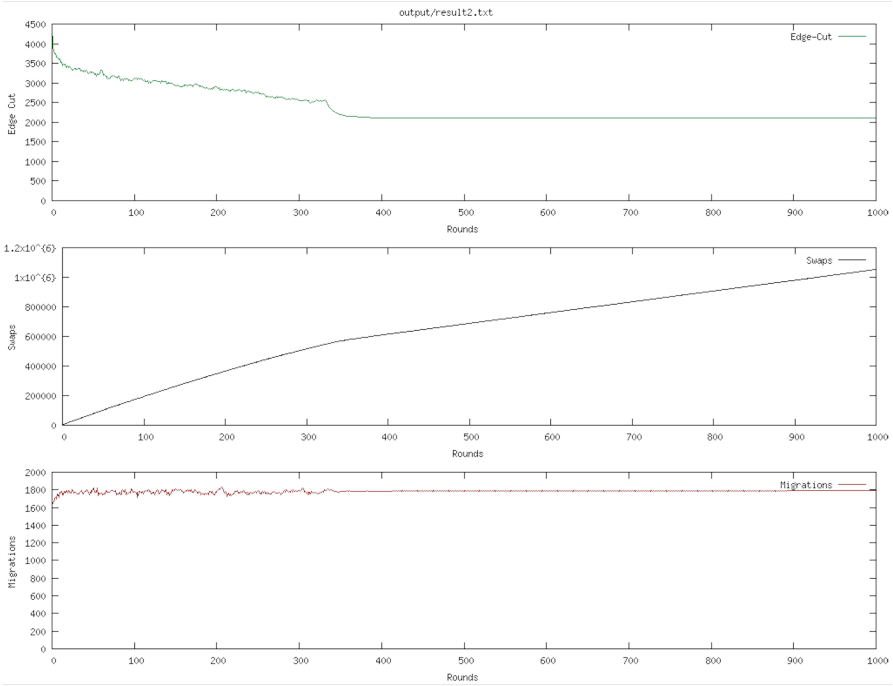

Figure 4: Results of the add20 graph.

```
2019-12-07 18:23:40 INFO  Jabeja:297 - round: 999, edge cut:2102, swaps: 1052366, migrations: 1790
Minimum edge cut: 2102.0
Mac:id2222-master francescostaccone$
```

Figure 5: Stats of the add20 graph.

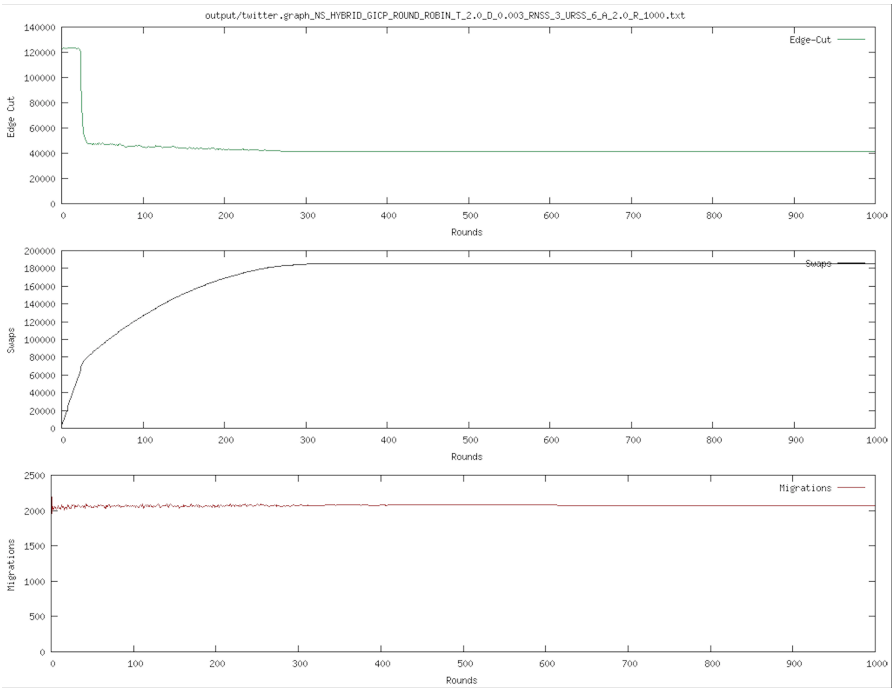

Figure 6: Results of the Twitter graph.

```
2019-12-07 18:39:17 INFO  Jabeja:297 - round: 999, edge cut:41312, swaps: 185499, migrations: 2075
Minimum edge cut: 41312.0
Mac:id2222-master francescostaccone$
```

Figure 7: Stats of the Twitter graph.

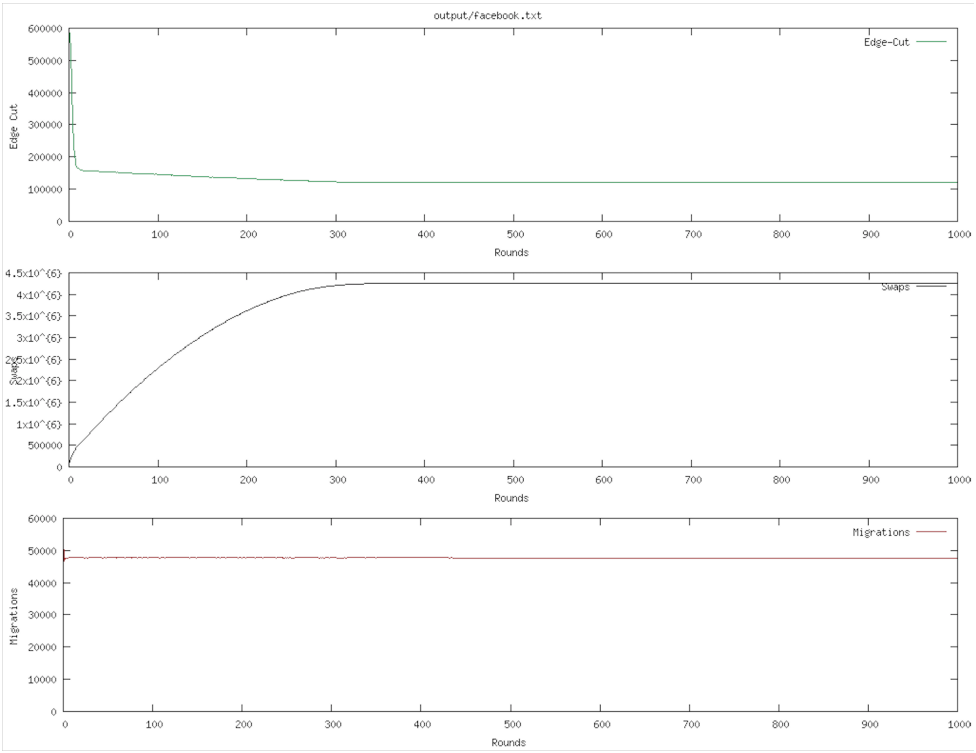

Figure 8: Results of the Facebook graph.

```
2019-12-07 19:04:37 INFO  Jabeja:297 - round: 999, edge cut:120424, swaps: 4248691, migrations: 47731
Minimum edge cut: 120424.0
Mac:id2222-master francescostaccone$
```

Figure 9: Stats of the Facebook graph.

Every result is obtained using the standard JA-BE-JA values: delta=0.003, rounds=1000, num_partitions=4, uniform_rand_sample_size=6, initial_T=2, randNeighborsSampleSize=3.

For **3elt** JA-BE-JA outputs 711 as minimum edge cut and reaches full convergence (no more swaps occure) in 501 rounds, although we can consider the algorithm converging after the 340 rounds. The number of swaps grows almost linearly during the first 200 rounds, then the growth gets smoother and after the first 500 rounds the number of swaps gets steady.

For **add20**, the minimum edge cut is 2102 and the number of swaps grows linearly during the first 400 rounds and reaches edge-cut convergence after 409 rounds, then the swaps growth gets smoother but it keeps increasing with a linear behaviour. We can notice that even if the algorithm already reached convergence (the edge cut does not change), the number of swaps keeps increasing. It means that some nodes are exchanging their color but there is no way to improve the number of inter-partition edges.

For **Twitter**, the minimum edge cut is 41312 and convergence is reached after the first 615 rounds. Moreover, we can notice that in 300 rounds the edge cut almost reaches convergence. The number of swaps grows linearly during the first 30 rounds, then the growth gets smoother until the first 300 rounds expires, at that point there is convergence.

For **Facebook**, the minimum edge cut is 120424 and convergence is reached after the first 552 rounds. Moreover, we can notice that in 300 rounds the edge cut almost reaches convergence. The number of swaps grows almost linearly during the first 200 rounds, then the growth gets smoother until the first 300 rounds expires, at that point there is convergence.

2.2 Task 2: tweaking JA-BE-JA configurations

We then implemented *simulated annealing*, where we also investigated the effect of restarting. This is mainly done while tweaking the configurations of JA-BE-JA, that is, the temperature T, the parameter of the energy function *alpha*, and *delta*.

Instead of cooling down linearly as was implemented in Task 1, using simulated annealing allows us to tweak parameters as mentioned above and use an *acceptance probability function* that decides whether the algorithm should keep the worse solution (with respect to the previous round). This is done in order to avoid the problem of getting stuck in a local maxima/minima. The acceptance probability takes the old cost (or energy), new cost and current T value and returns a value between 0 and 1 which is compared to a random number. And if the returned value is greater than the randomly generated one, the color swap happens. The function is given by:

$$AP = e^{\frac{(E_{new} - E_{old})}{T}}$$

First, we ran the algorithm with simulated annealing without restarts. The results are shown in Figures 10, 12 and 14 with their corresponding edge-cut, the number of swaps and migrations in Figures 11, 13 and 15, respectively for the **3elt**, **20add** and **Twitter** graphs.

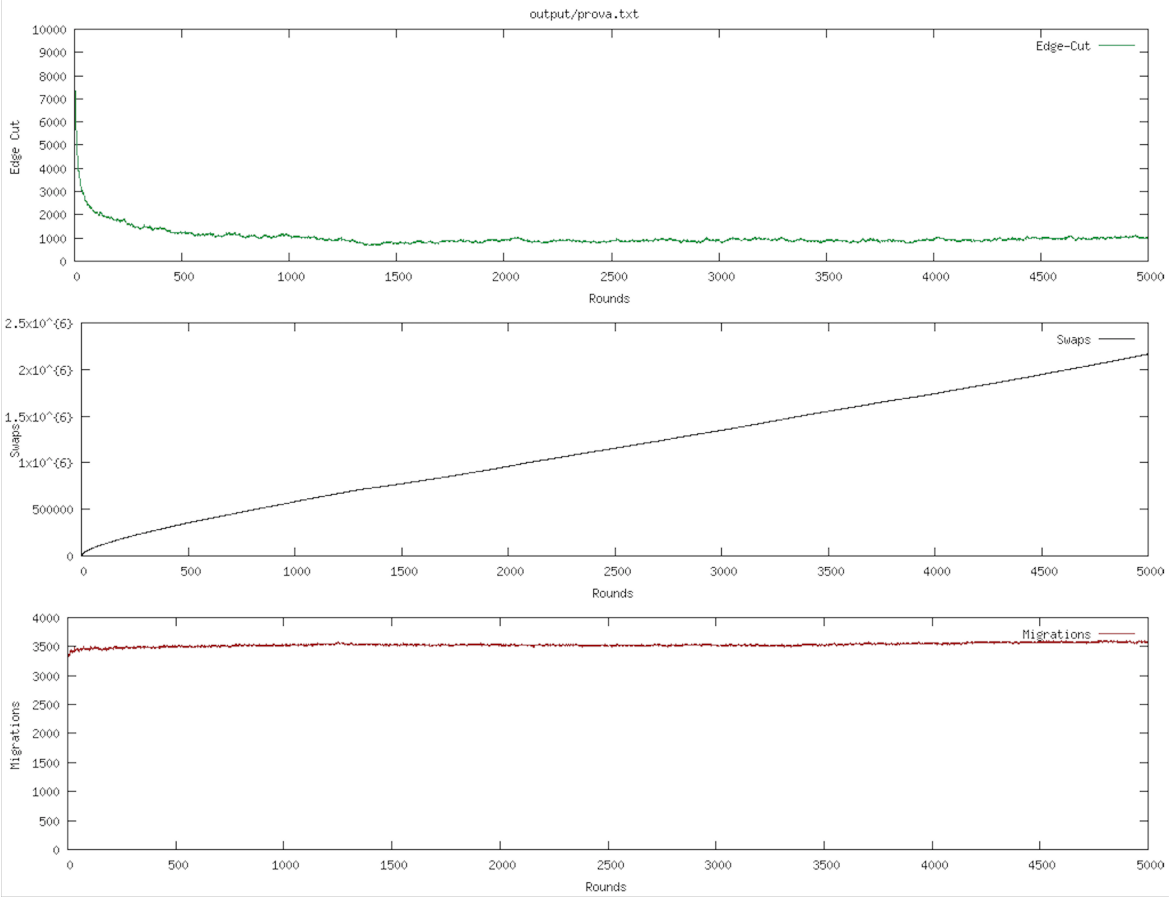

Figure 10: Results of the 3elt graph with simulated annealing without restarts.

```
2019-12-07 20:09:08 INFO  Jabeja:288 - round: 4999, edge cut:1017, swaps: 2163854, migrations: 3561
Minimum edge cut: 678.0
Mac:id2222-master francescostaccone$
```

Figure 11: Stats of the 3elt graph with simulated annealing without restarts.

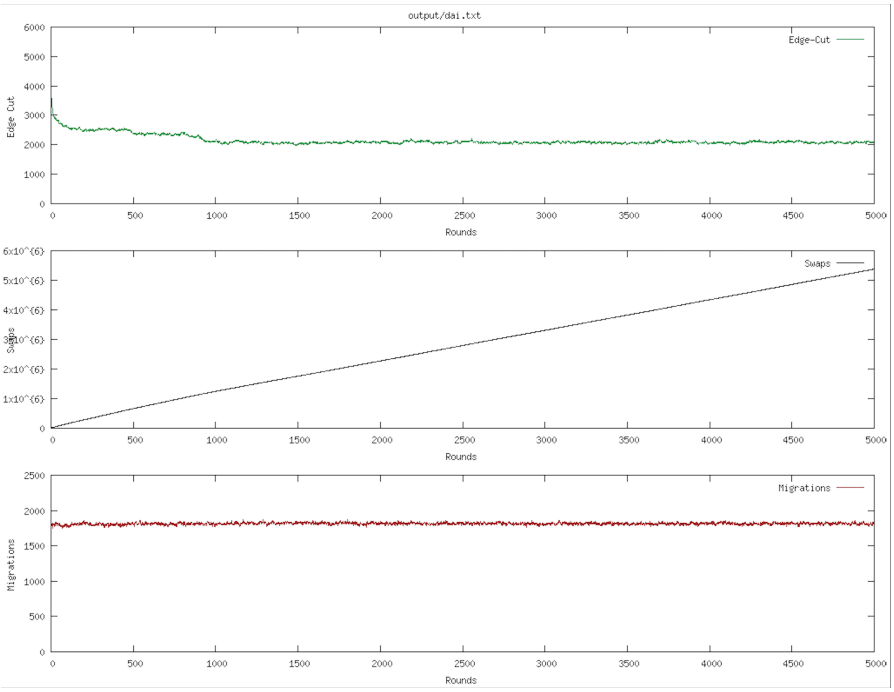

Figure 12: Results of the add20 graph with simulated annealing without restarts.

```
2019-12-07 19:45:02 INFO  Jabeja:286 - round: 4999, edge cut:2077, swaps: 5377079, migrations: 1829
Minimum edge cut: 1970.0
Mac:id2222-master francescostaccone$
```

Figure 13: Stats of the add20 graph with simulated annealing without restarts.

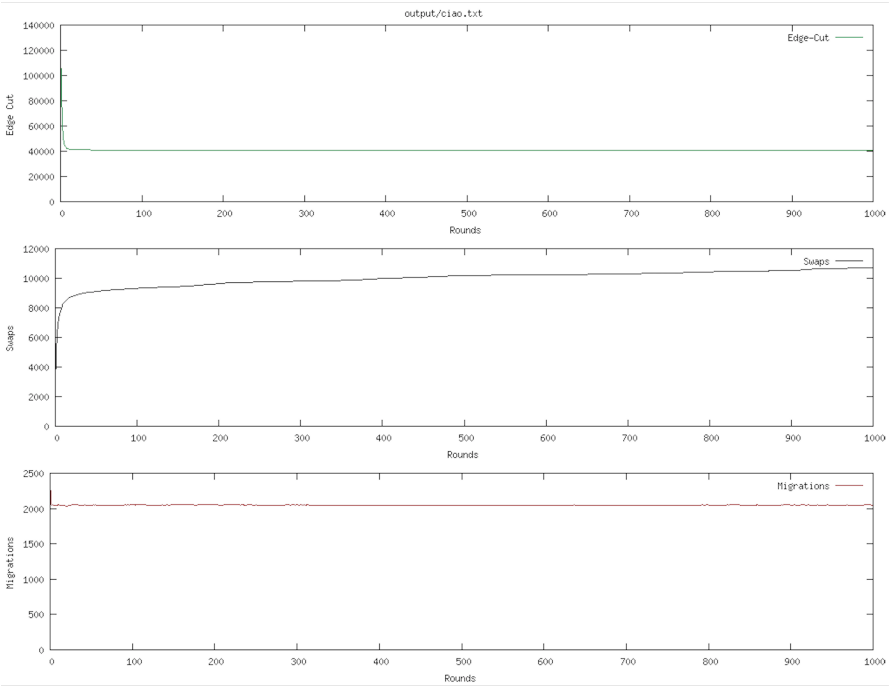

Figure 14: Results of the Twitter graph with simulated annealing without restarts.

```
2019-12-07 20:23:57 INFO  Jabeja:288 - round: 999, edge cut:40928, swaps: 10697, migrations: 2049
Minimum edge cut: 40924.0
Mac:id2222-master francescostaccone$
```

Figure 15: Stats of the Twitter graph with simulated annealing without restarts.

Here we analyzed the performances of JA-BE-JA, we tuned the hyperparameters to find the best set of parameters for each graph and identified possible improvements. Depending on the given graph, the configuration of the algorithm could slightly change. The above results are therefore obtained using different JA-BE-JA values:

- **3elt: delta=0.1, rounds=5000, initial_T=1, alpha= 2**
- **add20: delta=0.003, rounds=5000, initial_T=0.75, alpha= 0.9**
- **Twitter: delta=0.003, rounds=1000, initial_T=1, alpha= 2**

For **3elt** the minimum edge cut is 678 compared to the 711 of the standard case. Convergence is reached more or less after the first 2500 rounds. Moreover, we can notice that in 1000 rounds the edge cut almost reaches convergence. The number of swaps grows linearly since the acceptance probability function forces the exploration of also remote solutions.

For **add20**, the minimum edge cut is 1970 compared to the 2102 of the standard case. Convergence is reached more or less after the first 2500 rounds. Moreover, we can notice that in 1000 rounds the edge cut almost reaches convergence. The number of swaps grows linearly since the acceptance probability function forces the exploration of also remote solutions.

For **Twitter**, the minimum edge cut is 40924 compared to the 41312 of the standard case. Convergence is reached more or less after the first 40 rounds. The number of swaps grows rapidly and then almost linearly since the acceptance probability function forces use to keep exploring also remote solutions and the higher is the temperature parameter, the higher is the probability of swapping some nodes.

Now, to compare the results with simulated annealing with restarts, we show the results in Figures 16, 18 and 20 for the **3elt**, **20add** and **Twitter** graphs respectively, with the corresponding edge-cut, swaps and migrations in Figures 17, 19 and 21.

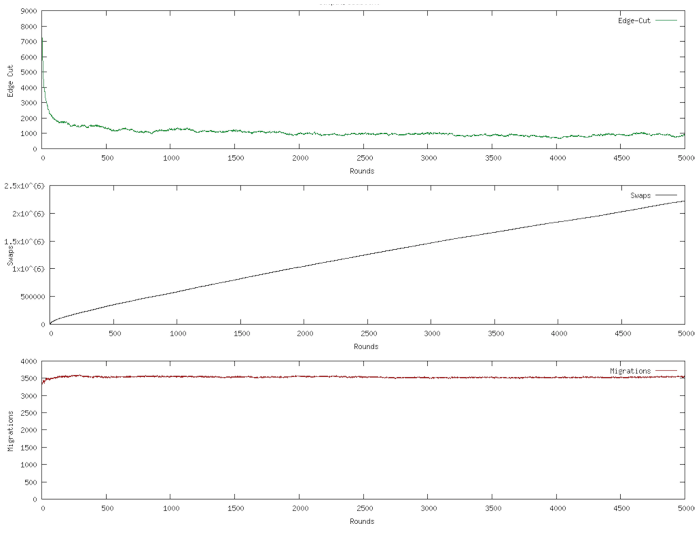

Figure 16: Results of the 3elt graph with restarting simulated annealing.

```
2019-12-08 00:24:27 INFO  Jabeja:288 - round: 4999, edge cut:923, swaps: 2222430, migrations: 3535
Minimum edge cut: 647.0
Mac:id2222-master francescostaccone$
```

Figure 17: Stats of the 3elt graph with restarting simulated annealing.

Here again we analyzed the performances of JA-BE-JA, we tuned the hyperparameters to find the best set of parameters for each graph and identified possible improvements. Depending

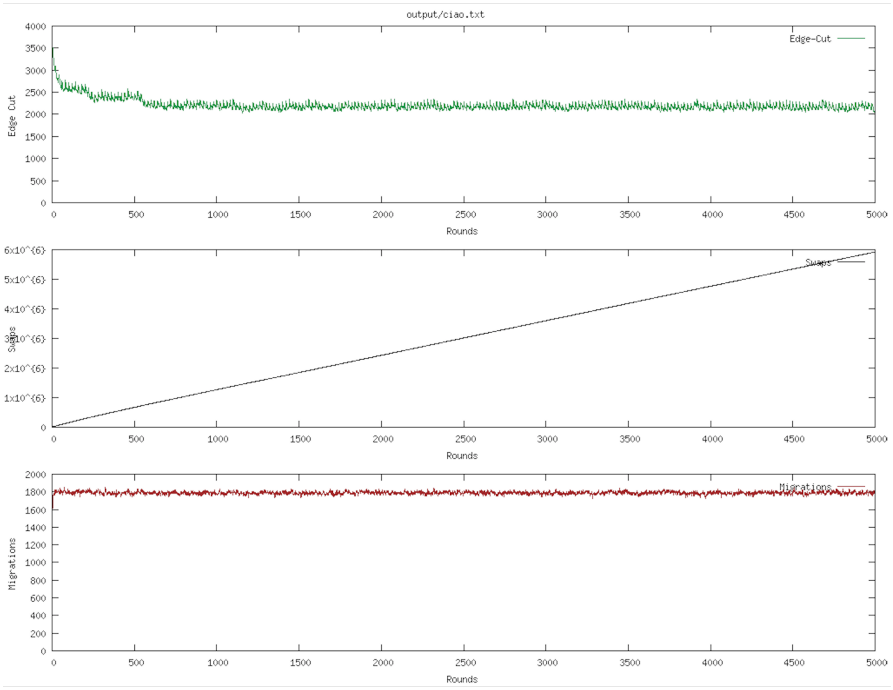

Figure 18: Results of the add20 graph with restarting simulated annealing.

```
2019-12-07 20:40:20 INFO  Jabeja:288 - round: 4999, edge cut:2022, swaps: 5303326, migrations: 1792
Minimum edge cut: 1943.0
Mac:id2222-master francescostaccone$
```

Figure 19: Stats of the add20 graph with restarting simulated annealing.

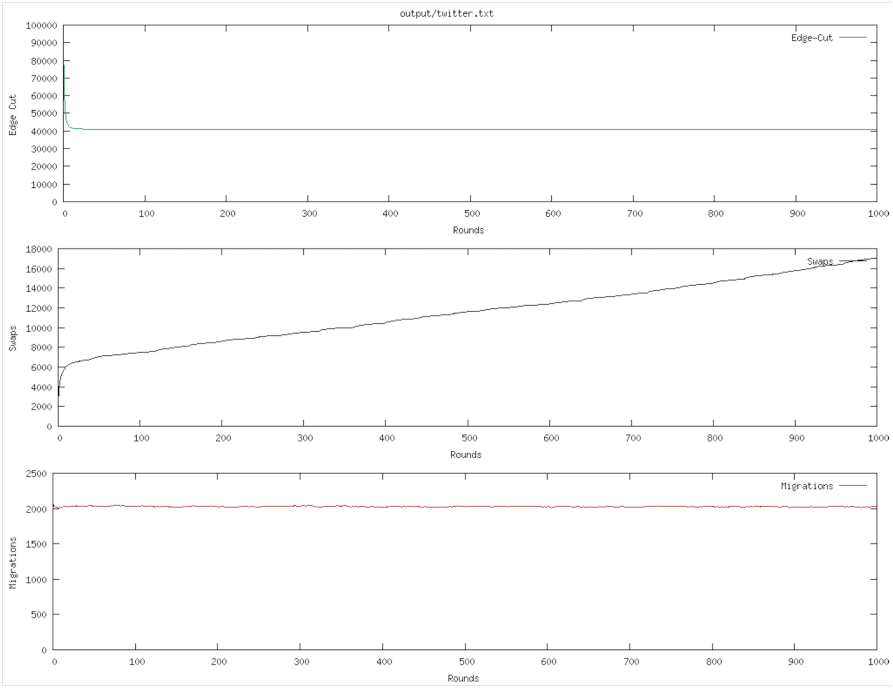

Figure 20: Results of the Twitter graph with restarting simulated annealing.

```
2019-12-07 20:31:14 INFO  Jabeja:288 - round: 999, edge cut:40862, swaps: 17027, migrations: 2029
Minimum edge cut: 40861.0
Mac:id2222-master francescostaccone$
```

Figure 21: Stats of the Twitter graph with restarting simulated annealing.

on the given graph, the configuration of the algorithm could slightly change. In addition to previous case, we measure after how many rounds the Temperature reaches the lower bound of 0.00001, and then wait for a number of rounds restart that is a multiple of that value, and represents a parameter of the algorithm that can be tuned. The above results are therefore obtained using different JA-BE-JA values:

- **3elt: delta=0.05, rounds=5000, initial_T=1, alpha= 2, restart=150**
- **add20: delta=0.03, rounds=5000, initial_T=0.75, alpha= 2, restart=50**
- **Twitter: delta=0.003, rounds=1000, initial_T=1, alpha= 0.9, restart=40**

For **3elt** the minimum edge cut is 647 compared to the 678 of the previous case. Convergence is reached more or less after the first 2500 rounds, with the periodic behavior given by the restart. The number of swaps grows linearly since the acceptance probability function forces the exploration of also remote solutions, the periodic behavior given by the restart.

For **add20**, the minimum edge cut is 1943 compared to the 1970 of the previous case. Convergence is reached more or less after the first 2500 rounds. Moreover, we can notice that in 1000 rounds the edge cut almost reaches convergence, with the periodic behavior given by the restart. The number of swaps grows linearly since the acceptance probability function forces the exploration of also remote solutions, the periodic behavior given by the restart.

For **Twitter**, the minimum edge cut is 40861 compared to the 40924 of the previous case. Convergence is reached more or less after the first 40 rounds, with the periodic behavior given by the restart. The number of swaps grows linearly since the acceptance probability function forces the exploration of also remote solutions, the periodic behavior given by the restart.

3 Optional task: defined probability function

Here, we define our own acceptance probability function and evaluate how using it the performance of the algorithm is affected. Like in the previous task, we compare the performance when simulated annealing is restarted and when not. We are using a Sigmoid function that takes in the same arguments as the previous acceptance probability function and is given by:

$$AP = \frac{1}{1 + e^{\frac{E_{new} - E_{old}}{T}}}$$

The results for when simulated annealing is not restarted are shown in Figures 22, 24 and 26, with the corresponding edge-cut, swaps and migrations are shown in Figures 23, 25 and 27 for the **3elt**, **20add** and **Twitter** graphs respectively.

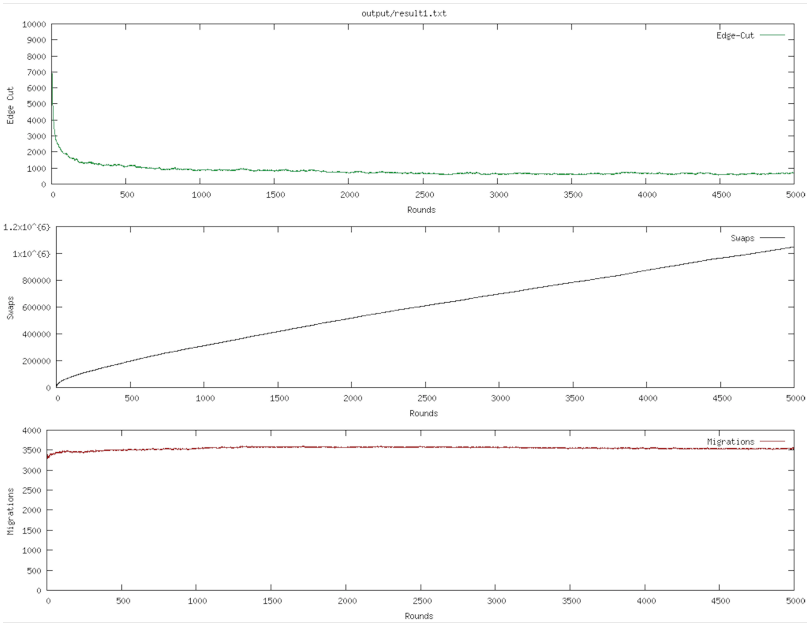

Figure 22: Results of the 3elt graph with Sigmoid probability function and without restarting simulated annealing.

```
2019-12-07 19:20:18 INFO  Jabeja:297 - round: 4999, edge cut:697, swaps: 1049133, migrations: 3538
Minimum edge cut: 564.0
Mac:id2222-master francescostaccone$
```

Figure 23: Stats of the 3elt graph with Sigmoid probability function and without restarting simulated annealing.

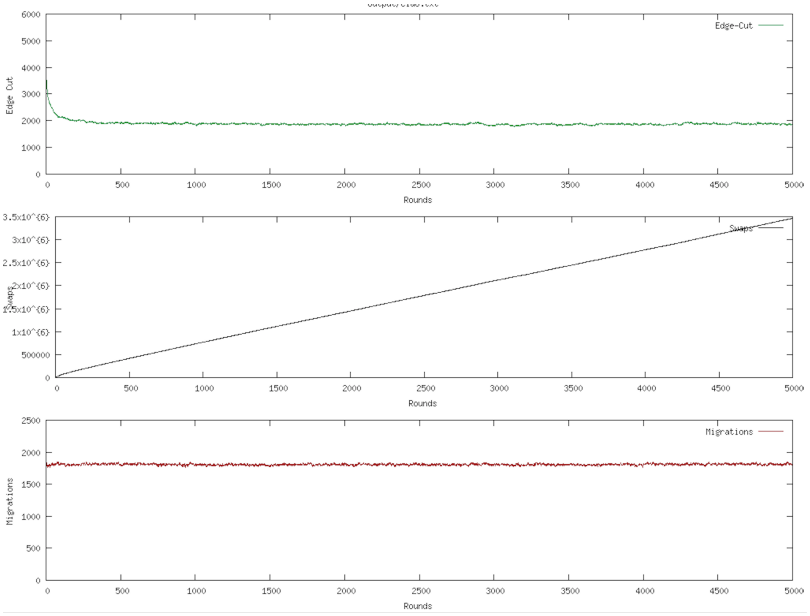

Figure 24: Results of the add20 graph with Sigmoid probability function and without restarting simulated annealing.

```
2019-12-08 00:34:46 INFO  Jabeja:289 - round: 4999, edge cut:1865, swaps: 3464310, migrations: 1813
Minimum edge cut: 1774.0
Mac:id2222-master francescostaccone$
```

Figure 25: Stats of the add20 graph with Sigmoid probability function and without restarting simulated annealing.

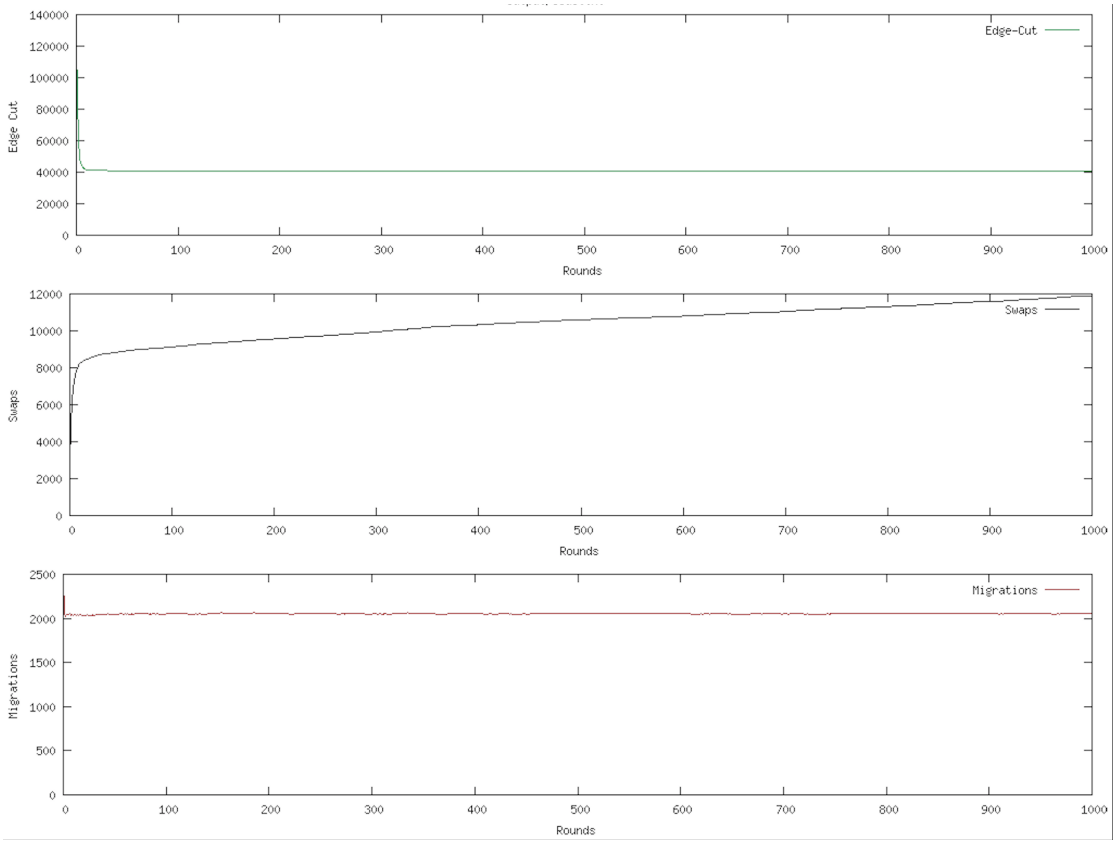

Figure 26: Results of the Facebook graph with Sigmoid probability function and without restarting simulated annealing.

```
2019-12-08 00:45:11 INFO  Jabeja:289 - round: 999, edge cut:40873, swaps: 11903, migrations: 2059
Minimum edge cut: 40867.0
Mac:id2222-master francescostaccone$
```

Figure 27: Stats of the Facebook graph with Sigmoid probability function and without restarting simulated annealing.

Here again we analyzed the performances of JA-BE-JA, we tuned the hyperparameters to find the best set of parameters for each graph and identified possible improvements. Depending on the given graph, the configuration of the algorithm could slightly change. The above results are therefore obtained using different JA-BE-JA values:

- **3elt: delta=0.003, rounds=5000, initial_T=1, alpha= 2**
- **add20: delta=0.03, rounds=5000, initial_T=0.75, alpha= 0.9**
- **Twitter: delta=0.1, rounds=1000, initial_T=1, alpha= 1**

For **3elt** the minimum edge cut is 564 compared to the 647 of the previous case. Convergence is reached more or less after the first 2500 rounds. Moreover, we can notice that in 1000 rounds the edge cut almost reaches convergence. The number of swaps grows linearly since the acceptance probability function forces the exploration of also remote solutions.

For **add20**, the minimum edge cut is 1774 compared to the 1943 of the previous case. Convergence is reached more or less after the first 3000 rounds. Moreover, we can notice that in 500 rounds the edge cut almost reaches convergence. The number of swaps grows linearly since the acceptance probability function forces the exploration of also remote solutions.

For **Twitter**, the minimum edge cut is 40867 compared to the 40861 of the previous (with

restarts) case. Convergence is reached more or less after the first 40 rounds. The number of swaps grows rapidly and then almost linearly since the acceptance probability function forces use to keep exploring also remote solutions and the higher is the temperature parameter, the higher is the probability of swapping some nodes.

And finally, the results with the simulated annealing restarted and with the Sigmoid probability function are shown in Figures 28, 30 and 32 for the **3elt**, **20add** and **Twitter** graphs respectively, with the corresponding edge-cuts, number of swaps and migrations shown in Figures 29, 31 and 33.

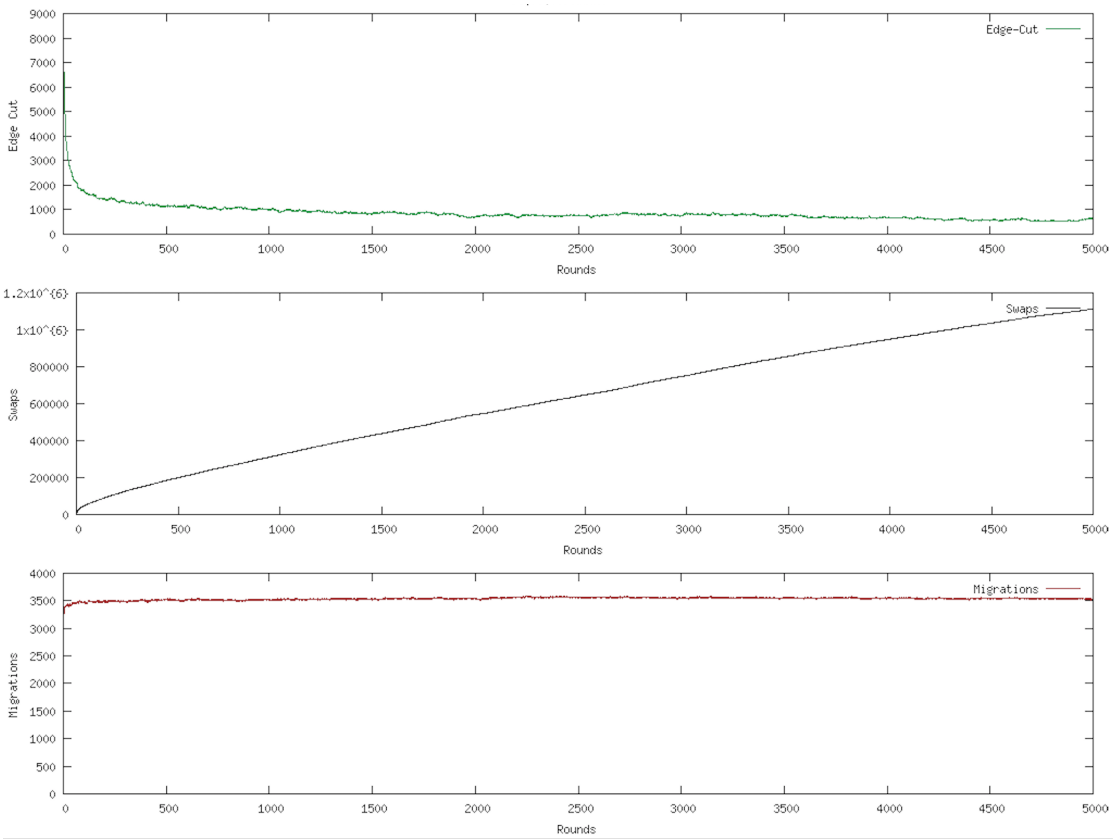

Figure 28: Results of the 3elt graph with Sigmoid probability function and with restarting simulate annealing.

```
2019-12-08 00:55:15 INFO  Jabeja:289 - round: 4999, edge cut:629, swaps: 1114585, migrations: 3541
Minimum edge cut: 513.0
Mac:id2222-master francescostaccone$
```

Figure 29: Stats of the 3elt graph with Sigmoid probability function and with restarting simulate annealing.

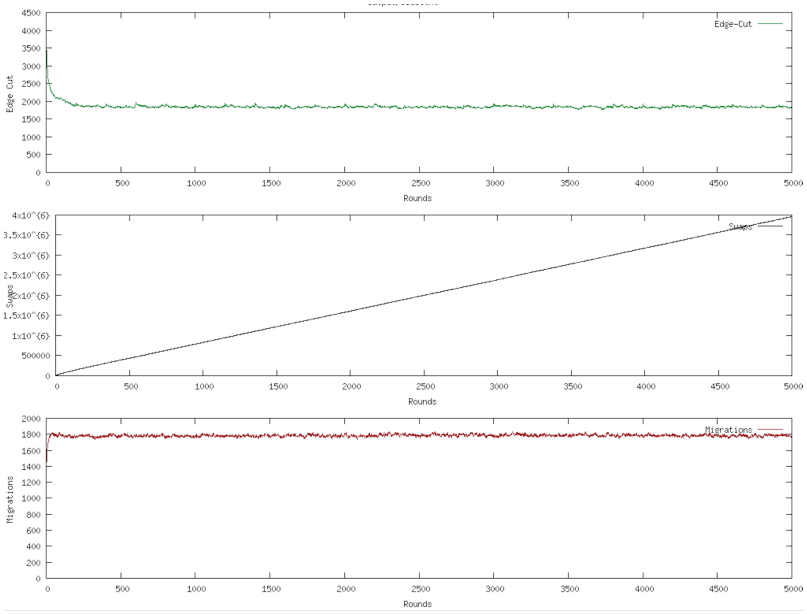

Figure 30: Results of the add20 graph with Sigmoid probability function and with restarting simulate annealing.

```
2019-12-08 01:12:29 INFO  Jabeja:289 - round: 4999, edge cut:1848, swaps: 3957832, migrations: 1794
Minimum edge cut: 1768.0
Mac:id2222-master francescostaccone$
```

Figure 31: Stats of the add20 graph with Sigmoid probability function and with restarting simulate annealing.

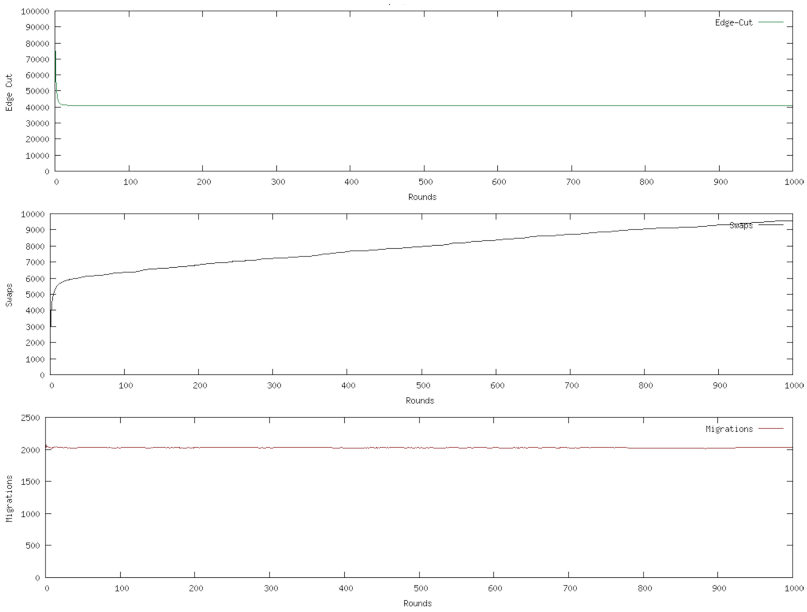

Figure 32: Results of the Twitter graph with Sigmoid probability function and with restarting simulate annealing.

```
2019-12-08 01:32:35 INFO  Jabeja:289 - round: 999, edge cut:40854, swaps: 9556, migrations: 2028
Minimum edge cut: 40852.0
Mac:id2222-master francescostaccone$
```

Figure 33: Stats of the Twitter graph with Sigmoid probability function and with restarting simulate annealing.

Here again we analyzed the performances of JA-BE-JA, we tuned the hyperparameters to find the best set of parameters for each graph and identified possible improvements. Depending on the given graph, the configuration of the algorithm could slightly change. In addition to previous case, we measure after how many rounds the Temperature reaches the lower bound of 0.00001, and then wait for a number of rounds restart that is a multiple of that value, and represents a parameter of the algorithm that can be tuned. The above results are therefore obtained using different JA-BE-JA values:

- **3elt: delta=0.05, rounds=5000, initial_T=1, alpha= 2, restart=150**
- **add20: delta=0.05, rounds=5000, initial_T=1, alpha= 1.5, restart=200**
- **Twitter: delta=0.1, rounds=1000, initial_T=1, alpha= 1.1, restart=40**

For **3elt** the minimum edge cut is 513 compared to the 564 of the previous case. Convergence is reached more or less after the first 2500 rounds, with the periodic behavior given by the restart. The number of swaps grows linearly since the acceptance probability function forces the exploration of also remote solutions, the periodic behavior given by the restart.

For **add20**, the minimum edge cut is 1768 compared to the 1774 of the previous case. Convergence is reached more or less after the first 2500 rounds. Moreover, we can notice that in 500 rounds the edge cut almost reaches convergence, with the periodic behavior given by the restart. The number of swaps grows linearly since the acceptance probability function forces the exploration of also remote solutions, the periodic behavior given by the restart.

For **Twitter**, the minimum edge cut is 40852 compared to the 40867 of the previous case. Convergence is reached more or less after the first 40 rounds, with the periodic behavior given by the restart. The number of swaps grows linearly since the acceptance probability function forces the exploration of also remote solutions, the periodic behavior given by the restart.